

Szenzorfüziós algoritmusok paramétereinek hatása a járműipari környezetérzékelés megbízhatóságára és a számításkapacitás igényre

Lindenmaier László* Aradi Szilárd*
Bécsi Tamás*

*Budapesti Műszaki és Gazdaságtudományi Egyetem, Közlekedés- és Járműirányítási Tanszék
1111 Budapest, Stoczek, Egry József u. 2. (e-mail: lindenmaier.laszlo@kjk.bme.hu).

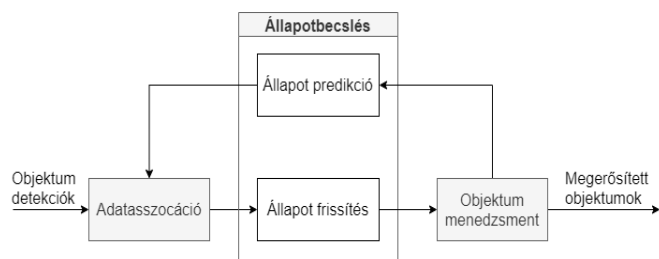
Abstrakt: A fejlett vezetéstámogató és autonóm rendszerek elterjedésében fokozott szerepe van a megbízható környezetérzékelésnek. A környezetérzékelés felé támasztott követelményeket jelenleg egyetlen járműipari szenzor sem képes önmagában kielégíteni. A szenzorfüzió célja, hogy az elérhető szenzoradatok felhasználásával pontosabb és megbízhatóbb modellt készítsen a környezetről, mintha bármelyik szenzort önmagában alkalmaznánk. A multi-object tracking algoritmusok architektúrája alkalmas több szenzorból származó objektum detekciók nyomon követésére és azok fuzionálására. A PDA és JPDA széleskörűen alkalmazott objektumkövető algoritmusok paramétereiben azonban jelentősen befolyásolhatják ezek számításkapacitás igényét és a fuzionált környezeti modell megbízhatóságát. Jelen tanulmány célja bemutatni ezen algoritmusok paraméter érzékenységét és ennek hatásait.

1. BEVEZETÉS

Az autonóm rendszerek elterjedésének egyik alapfeltétele az elvárt funkcionális biztonság, amelyben kitüntetett szerepe van a környezetérzékelésnek. Az önvezető járművek lehetséges hatásait több aspektusból is vizsgálták a kutatók (Földes & Csiszár, 2018) (Tettamanti, Varga, & Szalay, 2016) (Csiszár & Zarkeshev, 2017). Mivel egyetlen járműipari szenzor sem képes a funkcionális biztonság követelményeinek a kielégítésére, ezért a megbízható környezetérzékelés alapja a különböző szenzorok adatainak a fuzionálása (Zhu, Yuen, Mihaylova, & Leung, 2017) (Van Brummelen, O'Brien, Gruyer, & Najjaran, 2018). A szenzor adatok fuzionálása különböző absztrakciós szinteken valósulhat meg (Steinberg & Bowman, 2004). A fejlett vezetéstámogató rendszerekben a környezeti modell alapja gyakran a matematikai állapottal ellátott objektumok. Az ilyen környezeti modellalkotásnak első lépése az objektum detekció, amelyek nyomon követését ezután object tracking algoritmusok végzik. Az egyes szenzoroktól származó detekciók az érzékelés működési elvéből adódóan különböző előnyökkel és hátrányokkal rendelkeznek. Például a radarok nagy pontossággal képesek mérni a környező objektumok távolságát, azonban a szögfelbontásuk alacsony. Ezzel szemben a kamerák távolságmérése nem megbízható, de képesek az objektumokat klasszifikálni és sávhoz rendelni. A szenzoroktól származó objektumok fúziója két architektúra segítségével valósítható meg. A track-to-track fúzió a szenzorok detekcióinak nyomon követése után egyesíti a trackelt objektumokat. Míg a detekció szintű architektúrában a klasszikus objektumkövető (multi-object tracking) algoritmusok végzik az objektum detekciók fuzionálását is. Jelen tanulmányban ezen algoritmusok paramétereinek a hatását vizsgáljuk a környezetérzékelés megbízhatóságára és a számításkapacitás igényre fókuszálva.

2. OBJEKTUMKÖVETŐ ALGORITMUSOK

A szenzorok nyers adatainak feldolgozásából származó objektum detekciókat különböző bizonytalanságok terhelik. A szenzorok az objektumokat csak egy bizonyos valószínűséggel képesek detektálni, ezért előfordulhatnak kimaradó detekciók. Egy másik negatív jelenség a fals detekció, amely a valóságban nem létező, vagy irreleváns objektumot fed le. Továbbá, az objektumok állapotának megfigyelését zaj terheli és egyes állapotváltozókat nem képesek közvetlenül mérni a szenzorok (pl. a kamerák az objektumok sebességét). Az objektumkövető (multi-object tracking) algoritmusoknak két feladata van: meghatározni az időben változó objektumok számosságát, és becsülni az objektumok állapotát, ezzel mérsékelve a kimaradó- és fals detekciók hatását és szűrve a szenzorok mérési zaját. Az objektumkövető algoritmusok általános architektúrája (1. Ábra) három fő modulból áll: az adatasszociációból, az állapotbecslésből és az objektum menedzsmentből. Az algoritmus bemenetül a szenzoroktól származó objektum detekciók szolgálnak, míg a kimenete a létezőnek vélt, megerősített objektumok. Az adatasszociációt a 2.1, az állapotbecslést a 2.2, míg az objektum menedzsmentet a 2.3 fejezet részletezi.



1. Ábra: Az objektumkövető algoritmusok architektúrája

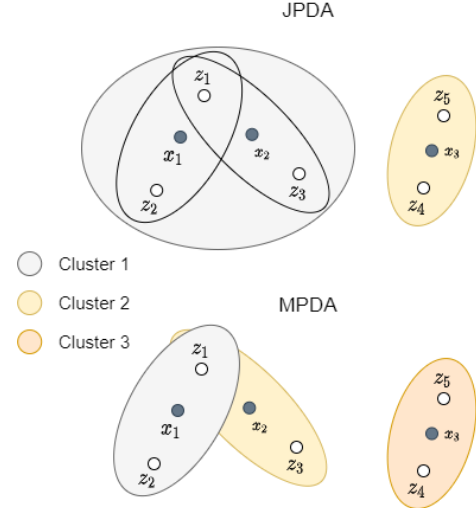
2.1 Adatasszociáció

Az adatasszociáció feladata, hogy meghatározza, hogy az objektum detekciók hogyan vesznek részt az egyes objektum állapotfrissítésében. Vagyis, az előző időpillanat objektumaihoz rendeli a szenzoroktól származó aktuális detekciókat. Eszerint megkülönböztetünk bináris és valószínűségi adatasszociációt. A bináris adatasszociáció kölcsönösen egyértelmű hozzárendeléssel közelíti meg az asszociációs problémát (Konstantinova, Udvarov, & Semerdjiev, 2003). Ezzel szemben a valószínűség alapú $A \in \{[0,1] \in \mathbb{R}^{N,M}\}$ asszociációs mátrix az alábbi szabály szerint kerül felépítésre:

$$a_{ij} \in A = p(x_i \leftrightarrow z_j), \quad (1)$$

ahol a $p(x_i \leftrightarrow z_j)$ annak az eseménynek a valószínűsége, hogy a z_j detekciót az x_i objektum indukálta, míg N és M az objektumok és a detekciók számosságát jelölik. A valószínűségi adatasszociáción alapuló objektumkövető algoritmusok széleskörű irodalommal rendelkeznek (Tugnait, 2004) (Garcia & de la Escalera, 2013) (Bar-Shalom, Daum, & Huang, The probabilistic data association filter, 2009). Jelen kutatás során a PDA (Probabilistic Data Association) (Bar-Shalom & Tse, Tracking in a cluttered environment with probabilistic data association, 1975) és a JPDA (Joint Probabilistic Data Association) (Bar-Shalom & Fortmann, Tracking and data association, 1988) algoritmusokat vizsgáltuk. A legnagyobb előnyük, hogy a $p(x_i \leftrightarrow z_j)$ asszociációs valószínűség kiszámítása során mindkét algoritmus figyelembe veszi, hogy x_i objektum detektálhatóságát és annak a valószínűségét, hogy a z_j detekció fals. A fő különbséget az jelenti, hogy a PDA algoritmus egy objektum nyomon követésére értelmezett. Kutatásunk során, ezért kiegészítettük, hogy alkalmazható legyen valóságos közúti scenáriókban is. Az adatasszociáció első lépése a valószínűtlen párok kiszűrése, mely egy d_g távolságküszöb meghatározásával történik. Amennyiben a $h(x_i)$ megfigyelési térre transzformált objektum állapot és a z_j detekció $d_{MH}(h(x_i), z_j)^2$ Mahalanobis-távolságának négyzete a d_g küszöbértéknél kisebb, x_i és z_j egy klaszterbe kerül. A JPDA algoritmus esetén, ha egy z_j detekció x_i és $x_i \neq x_n$ objektumhoz is tartozhat, a két objektum és az összes lehetséges detekciók egy klasztert alkotnak. A továbbfejlesztett MPDA (Multiple Probabilistic Data Association) algoritmus esetén egy klaszter legfeljebb egy objektumot tartalmazhat. Amennyiben egy z_j detekció több objektum küszöbtávolságán belül helyezkedik el, a hozzá legközelebbivel alkot egy klasztert. A két algoritmus klasztereinek különbségét a 2. Ábra szemlélteti. Az asszociációs valószínűségek kizárólag a klasztereken belül kerülnek kiszámításra, elhanyagolva ezzel a valószínűtlen párokat. A JPDA algoritmus komplexitása kombinatorikus egy klaszteren felül és a futásidőt a klaszterben lévő objektumok és detekciók száma befolyásolja. Ezzel szemben, az MPDA algoritmus komplexitása lineáris köszönhetően a megkötésnek, mely szerint egy klaszter legfeljebb egy

objektumot tartalmazhat. Érzékelhető tehát, hogy a d_g küszöbtávolság jelentősen befolyásolhatja a JPDA algoritmus számításkapacitás igényét. Míg az MPDA relaxáció hatással lehet az adatasszociáció megbízhatóságára, tekintve, hogy nem veszi figyelembe az objektumok egymásra hatását. Ezért, a kutatás során a d_g paraméter hatására fókuszáltunk.



2. Ábra: Klaszterezés a JPDA és az MPDA algoritmusban

2.2 Állapotbecslés

Az állapotbecslés célja, hogy a szenzorok méréseinek a zaját csökkentse és hogy az objektum állapot azon változóira becslést adjon, amelyeket a szenzorok nem képesek közvetlenül mérni. Az x állapot leírja az objektum pozícióját, sebességét és gyorsulását a járműhöz rögzített koordináta rendszerben a következők szerint:

$$x = [d_x \quad v_x \quad a_x \quad d_y \quad v_y \quad a_y], \quad (2)$$

ahol d_x , v_x , a_x a pozíció, a sebesség és gyorsulás longitudinális, míg d_y , v_y , a_y a laterális komponenseit jelölik. Az állapotbecslésre széleskörűen alkalmazott Bayes-féle rekurzív állapotbecslés elmélet (Chen, 2003) egyik speciális algoritmus a Kálmán-szűrő (Kalman, 1960). A rekurzív állapotbecslés első lépése az objektumok predikciója az aktuális időpillanatra, amelyet a Kálmán-szűrő a következő egyszerű lineáris transzformációval valósít meg:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}, \quad (3)$$

ahol $\hat{x}_{k-1|k-1}$ és $\hat{x}_{k|k-1}$ az x objektum $k-1$ és k időpillanatra vonatkozó a posteriori és a priori állapotbecslését jelölik. Az F_k átmeneti mátrix elemei egy egyszerű konstans gyorsulás modell szerint kerülnek meghatározásra:

$$F_k = I_2 \otimes \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

ahol Δt a k és a $k-1$ -edik időpillanat közt eltelt időt jelöli. Mivel a Kálmán-szűrő a modell zajokat Gauss eloszlásúnak feltételezi, a $\hat{P}_{k-1|k-1}$ állapotbecslés kovariancia prediktálása külön egyenletben kezelhető:

$$\hat{P}_{k|k-1} = F_k \hat{P}_{k-1|k-1} F_k^T + Q_k, \quad (5)$$

ahol Q_k a folyamat modell bizonytalanságát leíró kovariancia mátrixot jelöli, mely a $\sigma_{a,x}^2$ és $\sigma_{a,y}^2$ gyorsulás varianciákkal írható le:

$$Q_k = \begin{bmatrix} \sigma_{a,x}^2 q_k & \mathbf{0} \\ \mathbf{0} & \sigma_{a,y}^2 q_k \end{bmatrix}. \quad (6)$$

Míg q_k részmátrix a konstans gyorsulás modelltől vezethető le a következők szerint:

$$q_k = \omega \omega^T, \quad \omega = [\Delta t^2/2 \quad \Delta t \quad 1]^T \quad (7)$$

Akárcsak a predikció során, az állapotfrissítés is az a priori $\hat{x}_{k|k-1}$ állapotbecslés és $\hat{P}_{k|k-1}$ kovariancia frissítéséből áll. Az állapotbecslés a z_k szenzor detekció által kerül frissítésre az alábbiak szerint:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k \hat{x}_{k|k-1}), \quad (8)$$

ahol H_k a megfigyelési mátrix az $\hat{x}_{k|k-1}$ állapotbecslést a mérési térre transzformálja. A K_k Kálmán erősítés a $\hat{P}_{k|k-1}$ a priori kovariancia és az R_k mérési bizonytalanságot reprezentáló kovariancia alapján számítható:

$$K_k = \hat{P}_{k|k-1} H_k^T (H_k \hat{P}_{k|k-1} H_k^T + R_k)^{-1}. \quad (9)$$

Az állapotbecslés $\hat{P}_{k|k}$ frissített, a posteriori kovariancia mátrixa a következő egyenlet szerint kerül becslésre:

$$\hat{P}_{k|k} = (I - K_k H_k) \hat{P}_{k|k-1}. \quad (10)$$

2.3 Objektum menedzsment

Az objektum menedzsment feladata, hogy az eltűnő és a megjelenő objektumok kezelésével az objektumkövetést alkalmassá tegye valós környezetben történő felhasználásra. Az objektum menedzsment két megközelítés alapján történhet: a ciklus és a létezési valószínűség alapú. Mindkét megközelítésben azok a z_j detekciók, amelyek $\forall i: p(x_i \leftrightarrow z_j)$ asszociációs valószínűsége kisebb, mint az $1 - p_b$ küszöb egy új feltételes $x \in X_{init}$ objektumot inicializálnak. A ciklus alapú megközelítésben, egy feltételes objektum akkor kerül megerősítésre, amennyiben az N vizsgálati ciklusok során legalább M -szer megerősítésre kerülnek egy detekció által, vagyis az asszociációs valószínűségük meghalad egy p_c valószínűséget. Amennyiben pedig egy x_i objektumhoz P vizsgálati ciklusából Q -szor nem rendelhető hozzá egy $z_j \in Z$ detekció sem $p(x_i \leftrightarrow z_j) > p_c$ asszociációs valószínűséggel, az x_i törlésre kerül az objektum listából. Ezzel szemben, a létezési valószínűség alapú objektum menedzsment az x_i objektum $p(\exists x_i)$ létezési valószínűsége alapján kerül megerősítésre vagy törlésre. A PDA és a JPDA algoritmusok kiegészítésre kerültek a létezési valószínűség becslésével, így kialakítva az IPDA (Integrated Probabilistic Data Association) (Musicki, Evans, & Stankovic, Integrated probabilistic data association, 1994) és a JIPDA (Joint Integrated Probabilistic Data Association) (Musicki & Evans, Joint integrated probabilistic data association: JIPDA, 2004) algoritmusokat. Az x_i objektum $p(\exists x_i)$ létezési valószínűsége, mely Bernoulli

eloszlást követ, szintén Bayes-féle rekurzív elmélet szerint kerül becslésre. Ennek alapját az asszociációs valószínűségekből alkotott θ asszociációs hipotézisek képezik. Asszociációs hipotézisnek tekintjük az X objektum és Z detekciós halmaz összes elemének párosítását tartalmazó eseményt. Megjegyzendő, hogy az X halmaz kiegészítésre kerül a c , míg a Z halmaz a \mathbb{Z} és \emptyset speciális elemekkel, amelyek a fals detekciót és a nem létező és a nem detektálható objektumot jelölik. Ezek a speciális halmaz elemek a valós detekciókkal és objektumokkal ellentétben több halmaz elemhez is hozzárendelhetők. A θ asszociációs hipotézis valószínűsége a következő kifejezés szerint számítható:

$$p(\theta) = \prod_{e \in \theta} p(e), \quad (11)$$

ahol $e = \{x_i \leftrightarrow z_j\}$ egy objektum-detekció párt jelöl, beleértve a speciális halmaz elemeket is. Ezek alapján az x_i objektum a posteriori létezési valószínűsége az alábbi kifejezéssel számítható:

$$p_{k|k}(\exists x_i) = \frac{\sum_{\theta \in \theta_i^{\exists}} p(\theta)}{\sum_{\theta} p(\theta)}, \quad (12)$$

ahol θ_i^{\exists} azon asszociációs hipotézisek halmazát jelöli, amelyekben az x_i objektum létezőnek van feltételezve. Tekintve, hogy az MPDA adatasszociáció esetén a klaszterek legfeljebb egy objektumot tartalmazhatnak, a létezési valószínűség becslése így egyszerűsíthető:

$$p_{k|k}(\exists x_i) = \frac{p(x_i \leftrightarrow \emptyset) + \sum_{z \in C_i} p(x_i \leftrightarrow z)}{1 - p_{k|k-1}(\exists x_i)}, \quad (13)$$

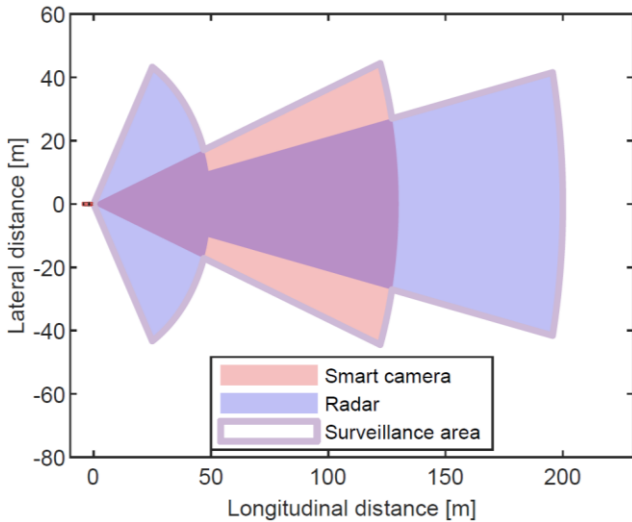
ahol $p(x_i \leftrightarrow \emptyset)$ az x_i objektum nem detektálhatóságának a valószínűségét és C_i a küszöbtávolságán belül található detekciók klaszterét jelöli. A $p_{k|k-1}(\exists x_i)$ prediktált létezési valószínűség a $p_s(x_i)$ túlélési valószínűség alapján a következőképpen számítható:

$$p_{k|k-1}(\exists x_i) = p_s(x_i) \cdot p_{k-1|k-1}(\exists x_i). \quad (14)$$

A rekurzív módon becsült létezési valószínűség alapján egy $x_i \in X_{init}$ feltételes objektum akkor kerül megerősítésre, ha az meghalad egy p_{init} küszöböt. Míg egy megerősített objektum $p_{k|k}(\exists x_i) < p_{del}$ létezési valószínűséggel automatikusan törlésre kerül az objektum listáról.

3. SENZORFÚZIÓ

A JIPDA és az MIPDA objektumkövető algoritmusok paramétereinek hatásvizsgálatát egy radarból és egy okos kamerából álló szenzorklaszteren hajtottuk végre, amelyet a 3. Ábra szemléltet. Annak érdekében, hogy az objektumkövető algoritmusok alkalmasak legyenek a két szenzor adatainak a fuzionálására, a szenzoradatok megfelelő kezelése szükséges. Ezt egy determinisztikus adatpufferen alapuló adat-szinkronizáció segítségével valósítottuk meg, amelynek célja, hogy az objektumkövető algoritmus megfelelő időben és sorrendben dolgozza fel a szenzorok adatait. A szenzorok adatai megadott sorrendben frissítik a közös objektumlistát, így az objektumkövető algoritmus fuzionálja a szenzorok adatait.

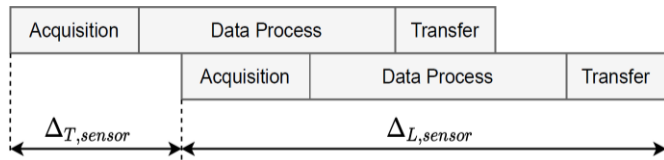


3. Ábra: A vizsgált szenzorok klaszter

A szenzorok adatait három okból szükséges szinkronizálni:

- különböző frekvenciával,
- aszinkron ütemezéssel,
- és nem megfelelő időrendben

küldik az adatokat. A cél, hogy megakadályozzuk, hogy egy szenzor adatai korábbi időpillanatra mutassanak, mint az objektumkövető által utoljára frissített adatok időbélyege. Ez ugyanis azt jelentené, hogy az állapotbecslés során múltba történő predikciót kéne végrehajtani. A szenzorok adatsomagjainak az életciklusát a 4. Ábra szemlélteti. A szenzoradat $\Delta_{L,sensor}$ teljes késése három részből tevődik össze: az adatgyűjtésből, az adatfeldolgozásból és az adattovábbításból. Az adatok maximális késése becsülhető a szenzor $\Delta_{T,sensor} = 1/f_{szenzor}$ periódus idejéből az $f_{szenzor}$ frekvencia alapján.

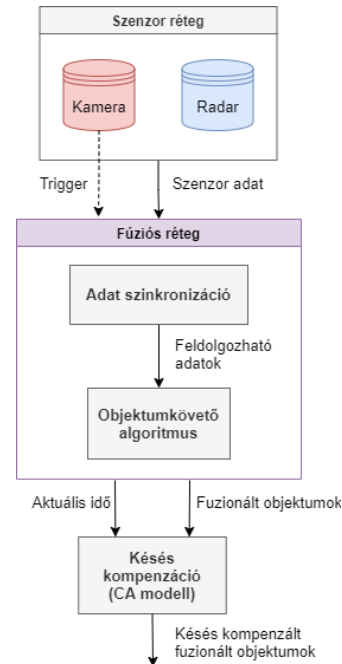


4. Ábra: A szenzoradat életciklusa (teljes késése)

Így egy szenzortól származó adatsomag akkor dolgozható fel, ha teljesül az alábbi két feltétel:

- egy adott időpillanatban az adatpufferben lévő minden adat feldolgozásra kerül, amelyek időbélyege régebbi, mint az aktuális időpillanat és bármelyik szenzor maximális késésének a különbsége
- ha egy szenzornak több adatsomagja is feldolgozható, csak a legfrissebb kerül felhasználásra

az összes többi szenzoradat az adatpufferbe kerül. Az így kialakult szenzorfüziós algoritmus architektúráját az 5. Ábra mutatja be. Miután az objektumkövető algoritmus frissítette az közös objektum listát, a posteriori késés az objektumok állapotának, a 2.2 fejezetben bemutatott konstans gyorsulás modellel történő predikcióval kerül kompenzálásra.



5. Ábra: A szenzorfüzió architektúrája

4. EREDMÉNYEK

A bemutatott szenzorfüziós algoritmusok paramétereinek a környezetérzékelés megbízhatóságára és a számítás kapacitás igényre gyakorolt hatását egy valós forgalmi körülmények között felvett mérésen vizsgáltuk. A 3. Ábra által szemléltetett szenzorok adatain felül egy szinkronban felvett digitális kamera képe került rögzítésre az M1/M7-es autópálya kivezető szakaszán. A kiértékeléshez használt szakasz összesen 7000 frame-ből áll (120 mp, ≈ 60 FPS) és összesen 23 objektum trajektóriáját tartalmazza. A valós, referencia objektumokat (ground truth) a digitális kamera képein végrehajtott annotáció útján állítottuk elő. A kalibrált kamera képén felvett referencia objektumokat perspektív transzformációval (homográfia) vetítettük le a jármű koordináta rendszerébe. A vetítés során figyelembe vettük az objektumok és a bounding box-ok méretét, ezzel szűrve a homográfia alapú transzformáció pontatlanságát. Végül az így kapott objektumok pozícióját 2D felülnézetben is ellenőriztük és korrigáltuk. A kiértékelés első lépése a referencia és a fuzionált objektumok közötti adatasszociáció, amelyet a Magyar-módszeren alapuló kölcsönösen egyértelmű hozzárendeléssel valósítottunk meg (Kuhn, 1955). A környezetérzékelés megbízhatóságát két aspektusból értékeltük: a fals pozitív és a fals negatív észlelések száma alapján, amelyekből egy $P \in \{[0,1] \in \mathbb{R}\}$ szerinti P_{pr} és P_{rc} úgynevezett precision és recall metrikák számíthatók (Kovács, Lindenmaier, Németh, Tihanyi, & Zarándy, 2018) a következőképpen:

$$P_{pr} = \frac{TP}{TP + FP}, \quad (15)$$

$$P_{rc} = \frac{TP}{TP + FN}, \quad (16)$$

ahol TP , FP és FN a valódi, a fals pozitív és a fals negatív észlelések teljes számát jelölik. A szenzorfüzió és a környezetérzékelés P_e eredő teljesítménye (megbízhatósága) a két metrika alapján a következő kifejezéssel számítható:

$$P_e = 2 \frac{P_{pr} P_{rc}}{P_{pr} + P_{rc}} \quad (15)$$

A kiértékelés másik fontos aspektusa a számításkapacitás igény, amelyet a komplexitás és a futásidő vizsgálatával végeztünk. Az objektumkövető algoritmusok $O(N, M)$ komplexitása az aktuálisan tárolt objektumok N és a szenzoradatcsomagban tárolt detekciók M számától függ. Így a komplexitás osztály vizsgálata során ezeket vettük figyelembe. Az algoritmusok futásidőjét a fúziós rétegben mértük (lásd 5. Ábra).

Ahogy az korábban említésre került, a JIPDA és a MIPDA algoritmusok teljesítményét és futásidőjét leginkább a d_g küszöbtávolság befolyásolja, melynek hatását 5 különböző értékkel teszteltük. Az küszöbtávolság vizsgált értékeit és a két szenzorfüziós algoritmus megbízhatósági metrikáira gyakorolt hatását az 1. és 2. Táblázat tartalmazza. A szenzorok által küldött szenzoradatok (objektum detekciók) precision, recall és P_e eredő metrikáit pedig a 3. Táblázat foglalja össze.

1. Táblázat A JIPDA algoritmus megbízhatósági metrikái

Paraméterek	P_{pr}	P_{rc}	P_e
$d_g = 10$	0,9212	0,8647	0,8921
$d_g = 40$	0,9355	0,8698	0,9015
$d_g = 70$	0,9417	0,8685	0,9036
$d_g = 100$	0,9419	0,8687	0,9038
$d_g = 130$	0,9350	0,8689	0,9007

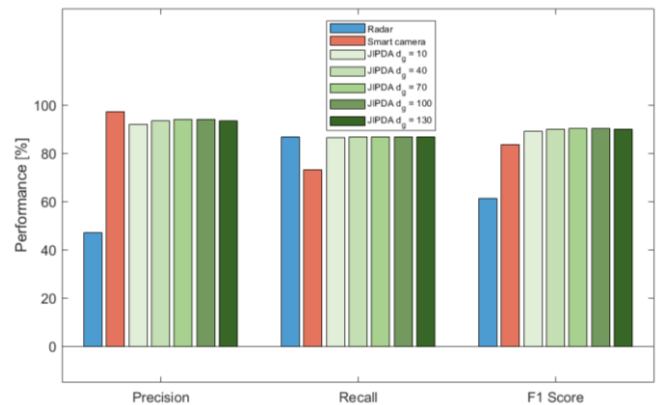
2. Táblázat A MIPDA algoritmus megbízhatósági metrikái

Paraméterek	P_{pr}	P_{rc}	P_e
$d_g = 10$	0,9091	0,8649	0,8864
$d_g = 40$	0,9423	0,8700	0,9047
$d_g = 70$	0,9486	0,8687	0,9069
$d_g = 100$	0,9488	0,8689	0,9071
$d_g = 130$	0,9493	0,8693	0,9075

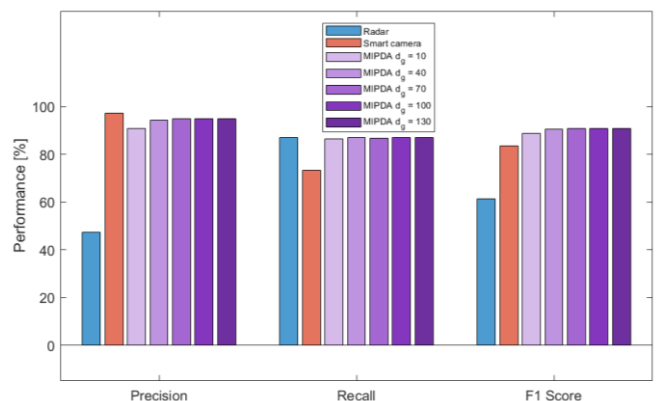
3. Táblázat A szenzoradatok megbízhatósági metrikái

Szenzorok	P_{pr}	P_{rc}	P_e
Kamera	0.9730	0.7327	0.8359
Radar	0.4724	0.8698	0.6122

Az eredmények alapján látható, hogy a d_g küszöbtávolság paraméter növelésével a szenzorfüziós algoritmusok megbízhatósága növekszik, de egy bizonyos szint fölött nem számottevően. Továbbá, a JIPDA algoritmus esetén a legnagyobb küszöbtávolság ($d_g = 130$) paraméterrel egy enyhe degradáció is tapasztalható, amely arra enged következtetni, hogy a küszöbtávolságot a megbízhatóság aspektusából sem érdemes minden határon túl növelni. A szenzorok metrikáival összehasonlítva elmondható, hogy mindkét szenzorfüziós algoritmus jelentősen javította a környezeti modell megbízhatóságát. A JIPDA objektumkövető algoritmuson alapuló szenzorfüzió optimális paraméterezése esetén, az eredő teljesítményt a legjobb szenzorhoz képest (kamera: 83,59 %) 6,79 %-kal javította, míg a MIPDA algoritmus esetén, ez az érték 7,16 % volt. Megfigyelhető, hogy az egyszerűbb MIPDA algoritmus a küszöbtávolság növelésével jobb eredményt produkált az összetettebb JIPDA algoritmushoz képest, és a vizsgált paraméterek esetén a teljesítmény is monoton növekvő tendenciát mutat a küszöbtávolság függvényében. A két algoritmus megbízhatósági metrikáit a 6. és 7. Ábra szemlélteti a vizsgált paraméterértékek függvényében a bemenetként szolgáló szenzorokkal összevetve.

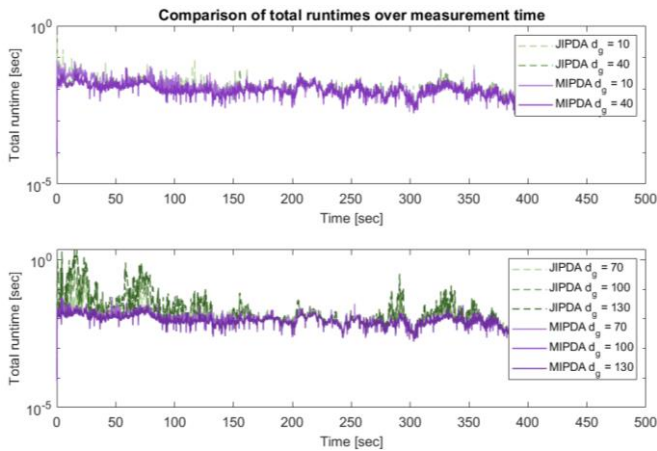


6. Ábra: A JIPDA algoritmusok alapuló szenzorfüzió megbízhatósági metrikái

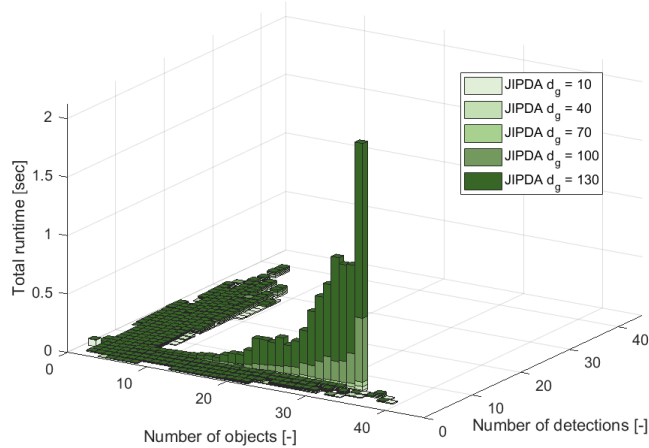


7. Ábra: A MIPDA algoritmusok alapuló szenzorfüzió megbízhatósági metrikái

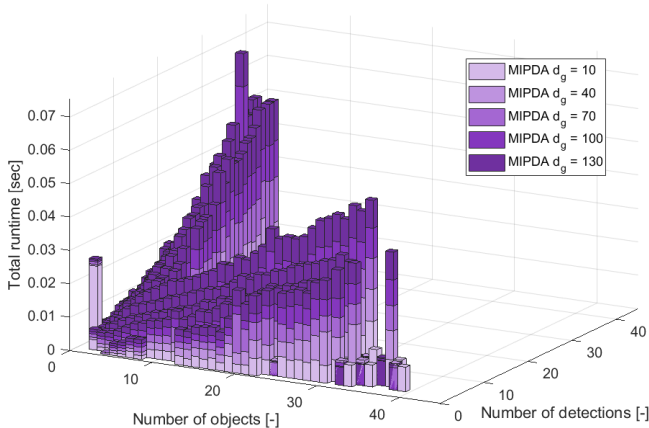
Bár a megbízhatósági metrikák esetén a különbség nem számottevő a két algoritmus között, komplexitásuk osztálya, így a futásidő karakterisztikák is eltérnek, különösen a küszöbtávolság növelésével.



8. Ábra: A szenzorfüziós algoritmusok futásideje az idő függvényében



9. Ábra: A JIPDA algoritmuson alapuló szenzorfüzió futásideje a tárolt objektumok és a bemeneti detekciók számának függvényében



10. Ábra: A MIPDA algoritmuson alapuló szenzorfüzió futásideje a tárolt objektumok és a bemeneti detekciók számának függvényében

Az algoritmusok futásidejének alakulását a 8. Ábra szemlélteti az idő függvényében, külön ábrázolva az alacsonyabb ($d_g \in \{10, 40\}$) és a magasabb ($d_g \in \{70, 100, 130\}$) küszöbtávolságok esetét. Látható, hogy míg az alacsonyabb küszöbtávolság értékek esetén az algoritmusok számításkapacitás igénye gyakorlatilag megegyezik, a küszöbtávolság növelésével jelentősen eltér a logaritmus skálán ábrázolt futásidő. Ezt a tendenciát erősíti a 9. és 10. Ábra által szemléltetett objektum és detekció számosság függvényében ábrázolt futásidő karakterisztika is, amely magyarázatot ad, a 8. Ábrán látható eltérésre is. Míg a MIPDA algoritmus esetén a tracking algoritmus által tárolt objektumok és a bemeneti detekciók számával lineárisan növekszik a futásidő, JIPDA algoritmusnál a számításkapacitás igény növekedése exponenciális tendenciát mutat. Ez annak köszönhető, hogy míg a MIPDA algoritmus klaszterei legfeljebb egy objektumot tartalmaznak, addig a JIPDA algoritmus esetén a nagy küszöbtávolságok és sok tárolt objektum esetén a klaszterek számos objektumból állhatnak. Mivel a θ asszociációs hipotézisek felépítése kombinatorikus, a sok objektumot tartalmazó klaszterek állapotbecslése és létezési valószínűség számítása kiugró számításkapacitást igényelnek.

5. KONKLÚZIÓ

Összességében elmondható tehát, hogy mindkét objektumkövető algoritmuson alapuló szenzorfüzió alkalmas a kiértékeléshez használt szenzorok és így a teljes környezetérzékelés megbízhatóságának a növelésére. Az optimális paraméterezés esetén mindkét algoritmus jelentősen növelte a szenzorok megbízhatósági metrikáit, amely a fejlett vezetéstámogató és önvezető rendszerekkel ellátott járművek környezetérzékelésének az alapja. Bár a megbízhatóság kapcsán nem tapasztalható szignifikáns változás, az algoritmusok számításkapacitás igényére nézve jelentős szerepe van vizsgált paramétereknek. Míg a MIPDA algoritmus komplexitása lineárisan függ a nyomon követett objektumok és a bemenetként szolgáló detekciók számától. Ezzel szemben a JIPDA algoritmus exponenciális növekedést mutat, amelynek a futásidőre gyakorolt hatása jelentős a nagy küszöbtávolságok esetén.

HIVATKOZÁSOK

- Bar-Shalom, Y., & Fortmann, T. (1988). *Tracking and data association*. Academic Press Boston.
- Bar-Shalom, Y., & Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, **Vol. 11**, 451-460.
- Bar-Shalom, Y., Daum, F., & Huang, J. (2009). The probabilistic data association filter. *IEEE Control Systems Magazine*, **Vol. 29**, 82-100.
- Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. *Statistics: A Journal of Theoretical and Applied Statistics*, **Vol. 182**, 1-69.
- Csiszár, C., & Zarkeshev, A. (2017). Demand-capacity coordination method in autonomous public transportation. *Transportation research periodica*, **Vol. 27**, 784-790.

- Földes, D., & Csiszár, C. (2018). Framework for planning the mobility service based on autonomous vehicles. *Smart City Symposium (SCSP)* (old.: 1-6). Prague: IEEE.
- Garcia, F., & de la Escalera, A. (2013). Joint Probabilistic Data Association fusion approach for pedestrian detection. *IEEE Intelligent Vehicles Symposium (IV)* (old.: 1344-1349). IEEE.
- Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, **Vol. 82**, 35-45.
- Konstantinova, P., Udvarov, A., & Semerdjiev, T. (2003). A study of a target tracking algorithm using global nearest neighbor approach. *Proceedings of the International Conference on Computer Systems and Technologies*, (old.: 290-295).
- Kovács, L., Lindenmaier, L., Németh, H., Tihanyi, V., & Zarándy, Á. (2018). Performance Evaluation of a Track to Track Sensor Fusion Algorithm. *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications* (old.: 1-2). Budapest: VDE.
- Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, **Vol. 2**, 83-97.
- Musicki, D., & Evans, R. (2004). Joint integrated probabilistic data association: JIPDA. *IEEE Transactions on Aerospace and Electronic Systems*, **Vol. 40**, 1093-1099.
- Musicki, D., Evans, R., & Stankovic, S. (1994). Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, **Vol. 39**, 1237-1241.
- Steinberg, A., & Bowman, C. (2004). Rethinking the JDL data fusion levels. *Nssdf Jhapl*, **Vol. 38**, 39.
- Tettamanti, T., Varga, I., & Szalay, Z. (2016). Impacts of autonomous cars from traffic engineering perspective. *Periodica Polytechnica Transportation Engineering*, **Vol. 44**, 244-250.
- Tugnait, J. (2004). Tracking of multiple maneuvering targets in clutter using multiple sensors, IMM, and JPDA coupled filtering. *IEEE Transactions on Aerospace and Electronic Systems*, **Vol. 40**, 320-330.
- Van Brummelen, J., O'Brien, M., Gruyer, D., & Najjaran, H. (2018). Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Engineering Technologies*, **Vol. 89**, 384-406.
- Zhu, H., Yuen, K.-V., Mihaylova, L., & Leung, H. (2017). Overview of Environment Perception for Intelligent Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, **Vol. 18**, 2584-2601.